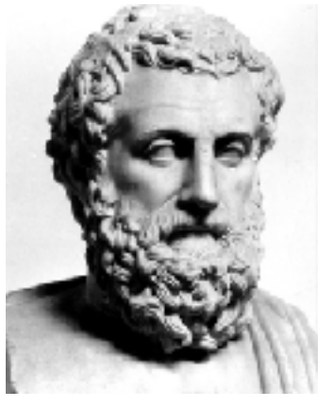
Artificial Intelligence For NLP Lesson-05

人工智能与自然语言处理课程组

2019.Aug. 03



- TF-IDF Keywords
- Words Cloud
- Based on Graph and Word Embedding
- Text-Rank (We will talk in future)
- Based on Machine Learning(We will talk in future)

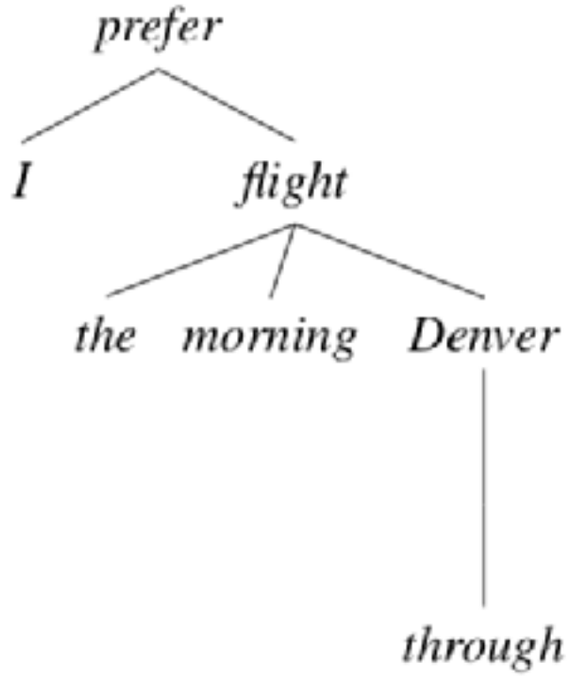
- (on line coding presentation)

Keywords and Words-cloud

NER

DEPENDENCY PARSING

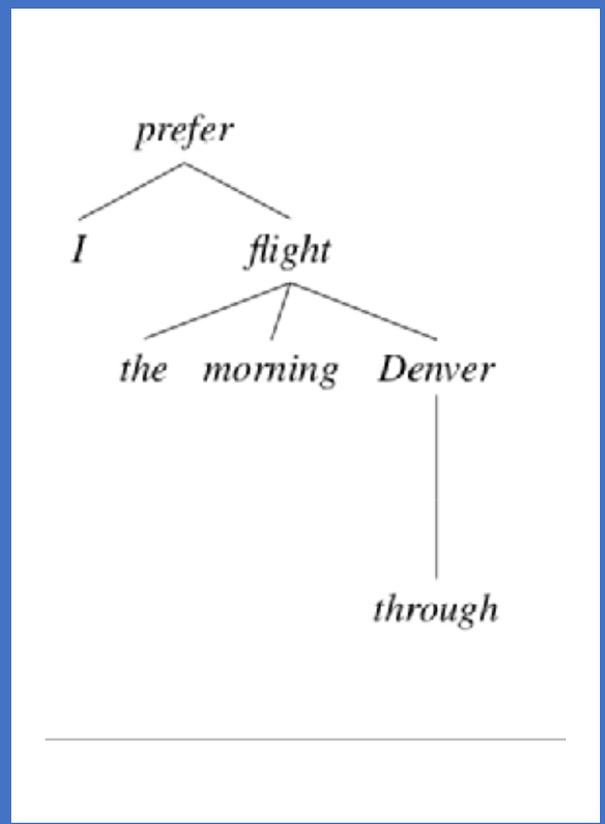
BOOLEAN SEARCH



Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

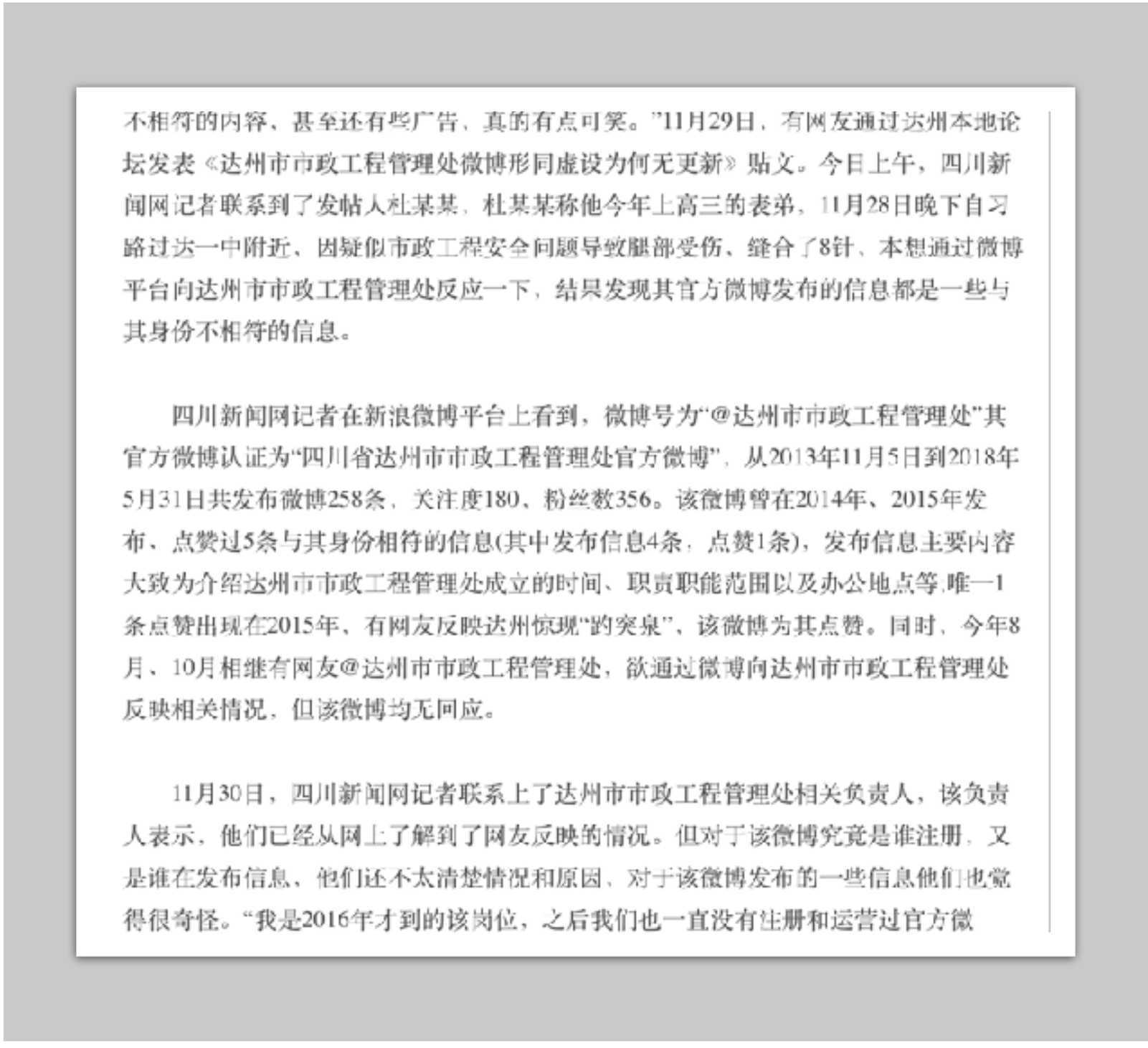
Dependency Parsing

Natural Language Processing: Chapter-14, this book is saved on our github repository



Project - 01

- Extracting the Person's talk from New Corpus.
  - Dataset: News Corpus
  - Toolset: Pandas, Matplotlib, Numpy, Jieba, Gensim
  - Applicatoin:
    - Trending Analysis
    - Knowledge Graph
    - Semantic Analysis
    - Risk Predication



不相符的内容，甚至还有些广告，真的有点可笑。”11月29日，有网友通过达州本地论坛发表《达州市市政工程管理处微博形同虚设为何无更新》贴文。今日上午，四川新闻网记者联系到了发帖人杜某某，杜某某称他今年上高三的表弟，11月28日晚下自习路过达一中附近，因疑似市政工程安全问题导致腿部受伤，缝合了8针，本想通过微博平台向达州市市政工程管理处反应一下，结果发现其官方微博发布的信息都是一些与其身份不相符的信息。

四川新闻网记者在新浪微博平台上看到，微博号为“@达州市市政工程管理处”其官方微博认证为“四川省达州市市政工程管理处官方微博”，从2013年11月5日到2018年5月31日共发布微博258条，关注度180，粉丝数356。该微博曾在2014年、2015年发布、点赞过5条与其身份相符的信息(其中发布信息4条，点赞1条)，发布信息主要内容大致为介绍达州市市政工程管理处成立的时间、职责职能范围以及办公地点等，唯一1条点赞出现在2015年，有网友反映达州惊现“跳突泉”，该微博为其点赞。同时，今年8月、10月相继有网友@达州市市政工程管理处，欲通过微博向达州市市政工程管理处反映相关情况，但该微博均无回应。

11月30日，四川新闻网记者联系上了达州市市政工程管理处相关负责人，该负责人表示，他们已经从网上了解到了网友反映的情况。但对于该微博究竟是谁注册，又是谁在发布信息，他们还不太清楚情况和原因，对于该微博发布的一些信息他们也觉得很奇怪。“我是2016年才到的该岗位，之后我们也一直没有注册和运营过官方微

Previous Remain

- 1. How to get related words by word2vec?

1. Keywords

Which Words are important?

金正男遇害案成悬案?最后一名嫌犯越南籍女子获释

2019-05-03 11:04:10 来源: 东方网

1113

易信

微信

QQ空间

微博

（原标题：马来西亚释放"谋杀金姓男子"越南女嫌犯）

历经了三年的曲折，世界上最引人注目的谋杀谜团之一却匆匆收场。

据俄联社报道，今天（5月3日）上午7时20分左右，被指控杀害朝鲜最高领导人金正恩同父异母兄弟金正男的第二名女性——越南公民段氏香从马来西亚一所女子监狱出狱。

2. TF-IDF

- Term Frequency – Inverse Document Frequency
  - The Simplest approach is to assign the weight to be equal to the number of occurrences of term *t* in document *d*. → *Term Frequency (tf)*
- It is more commonplace to use *document frequency df*, defined to be the number of documents in the collection that contain term *t*.
- Denoting as usual the total number of documents in a collection by *N*, we define the *inverse document frequency* (idf) of a term t as follows:

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t.$$

In other words, tf–idf*t,d* assigns to term *t* a weight in document *d* that is

highest when *t* occurs many times within a small number of documents (thus lending high discriminating power to those documents);

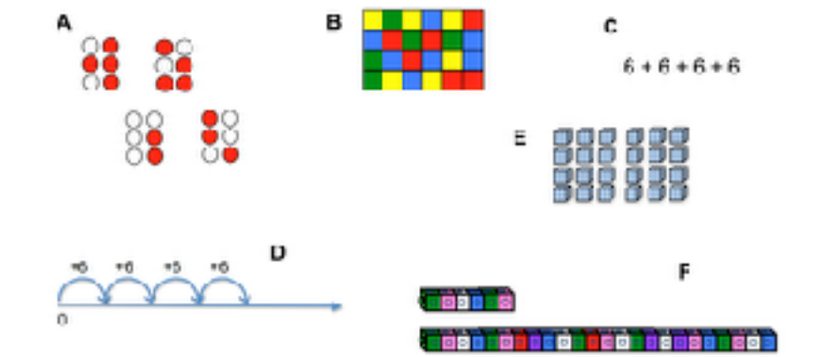
lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal);

lowest when the term occurs in virtually all documents.

- (online-coding for *tf-idf* and *word cloud*)

The Vector Space model for scoring

- As we mentioned in Lesson-5, Word2Vec, and in this lesson TFIDF. The representation of a set of documents as vectors in a common vector space is known as the *vector space model* and is fundamental to a host of information retrieval (IR) operations including scoring documents on *a query, document classification, and document clustering*. We first develop the basic ideas underlying vector space scoring; a pivotal step in this development is the view of queries as vector.



The importance of Representation

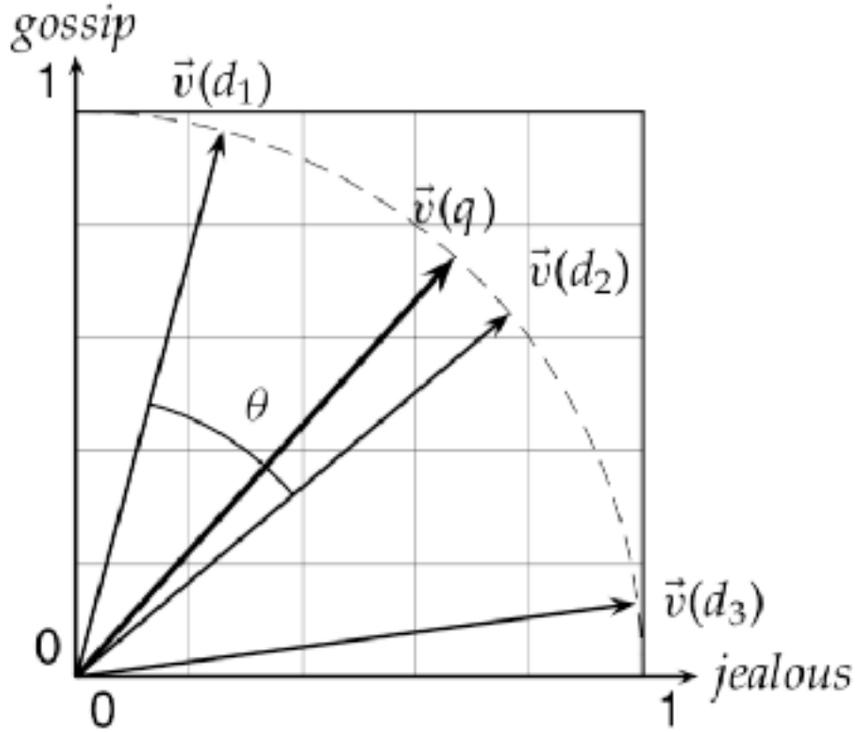
- Representation + Policy

Scikit-Learning TFIDF and Simplest Classification Model

- (on-line coding using scikit learning)

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|},$$

Cosine similarity illustrated: $\text{sim}(d_1, d_2) = \cos\theta$.



Boolean Search

1. To Process large document collections quickly

2. To allow more flexible matching operations

3. To allow ranked retrieval.

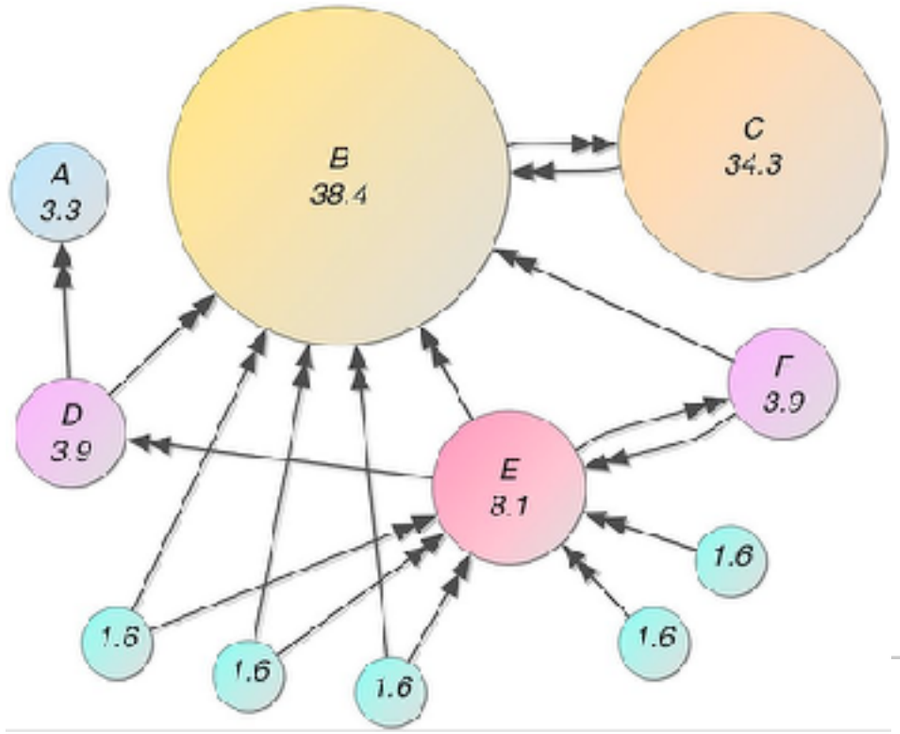
	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0
...						

$$110100 \text{ AND } 110111 \text{ AND } 101111 = 100100$$

Ranking Using TFIDF

- With TFIDF we could build a search engine.

PageRank



Iterative [edit]

At $t = 0$, an initial probability distribution is assumed, usually

$$PR(p_i;0) = \frac{1}{N}.$$

where N is the total number of pages, and $p_i;0$ is page i at time 0.

At each time step, the computation, as detailed above, yields

$$PR(p_i;t+1) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j;t)}{L(p_j)}$$